Architecting in a Solution Costing Context

Early Experiences with Solution-Based Estimating

Eltjo Poort
Solution Architect, CGI
Rotterdam, The Netherlands
eltjo.poort@cgi.com

Eric van der Vliet
EAS – Estimation Center CGI
Rotterdam, The Netherlands
eric.van.der.vliet@cgi.com

***Abstract*—Organizations require cost estimates for delivering and operating software-intensive solutions. One of the factors impacting these costs is the solution's architecture. Applying solution architecting practices helps improve the accuracy of early cost estimating. In CGI, we have started to emphasize the application of architecting practices in cost estimating, and this paper reports on the early experiences of this initiative, which we call Solution-Based Estimating. We present an architectural viewpoint designed to address the costing concern, and report some early feedback from projects that have applied the Solution-Based Estimating approach.**

Keywords— software architecture; solution architecture; software estimating; software project management

I. INTRODUCTION

Before a project to deliver a software-intensive solution can start, organizations need a cost estimate to determine the project budget. Similarly, decisions to develop, acquire and operate software-intensive systems or IT-services are generally based on a business case in which a trade-off is made between the benefits and the investment required. A large body of knowledge exists regarding cost estimation of software systems [1] [2]; however, the role of architecture in estimating has so far received relatively little attention. CMMI identifies in a sub-practice that estimates should be based on the product architecture, but does not give an guidance as to how this should be achieved [3]. Since research shows that applying solution architecture practices is indeed correlated with higher accuracy of budget prediction [4], it is important to gain more insight into the relationship between architecture and estimation. Organizations that supply or make use of software-intensive solutions could benefit from such insights.

As we have previously argued [5], the role of architecture in estimating becomes especially important for software-intensive systems and services when they consist of multiple dissimilar components; we call such systems and services *heterogeneous IT-based solutions*, or simply *heterogeneous solutions*. Such enterprise-level solutions can consist of a wide variety of components, such as custom-made software, embedded software, commercial Off-The Shelf (COTS) software (e.g. platform and utility software, Operating Systems, databases, middleware, enterprise applications), IT infrastructure (e.g. servers, storage, network elements), services to manage IT components or to perform business processes and/or organizational entities that deliver such services.

Delivering such enterprise-level, heterogeneous solutions to clients is the core business of CGI[1], and improving the accuracy of cost estimation for such solutions is a key business concern for IT Services providers like CGI. This concern is especially urgent in the early stages of solution design, when the architecture is the only representation of the solution's inner workings, because no detailed design exists yet at that stage. For this reason, the organization has started an initiative to consistently involve architecture in solution delivery estimates, called Solution-Based Estimating. Our Solution-Based Estimating approach is based on strong involvement of the solution architect in the estimating process, and the presence of a documented solution architecture as input to the estimate.

Solution-Based Estimating is part of the Risk- and Cost-Driven Architecture (RCDA) approach [6]. RCDA is a solution architecting approach that places strong focus on the economic context of architectures. Its first principle is that risk and cost drive architectural decision making, and Solution-Based Estimating is one of the implementations of that principle.

This experience report is structured as follows. We have previously published earlier results from the Solution-Based Estimating initiative in [5]. For the convenience of the reader, we will briefly summarize these results. We will then proceed to present the new contributions of this short paper:

- an architectural viewpoint designed to address stakeholder concerns associated with cost estimation, called the delivery breakdown viewpoint;
- early feedback from projects that have applied the approach.

Finally, we will summarize our conclusions.

II. BACKGROUND

In this section, we summarize the main results presented in [5] for the reader's convenience: the solution breakdown structure and the Solution-Based Estimating process.

[1] CGI is a global IT and business process services provider delivering business consulting, systems integration and outsourcing services

doi: 10.1109/WICSA.2015.9

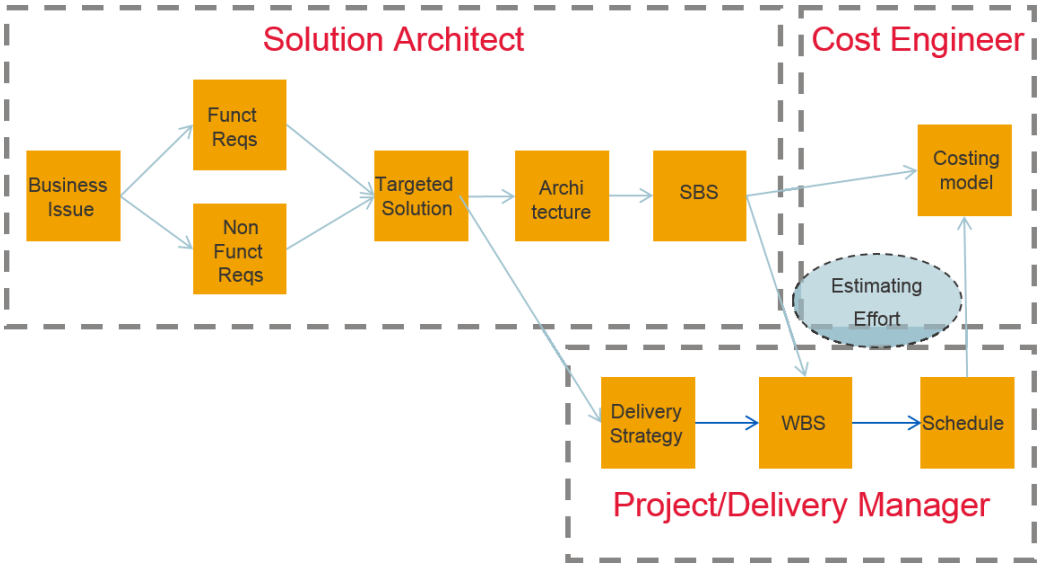

Figure 1Artifacts and roles in solution-based estimating (from [5])

A. *Solution breakdown structure*

In a solution breakdown structure (SBS), the solution is broken down into deliverable elements, usually visualized as an inverted tree diagram. This breakdown shows the costing structure of the solution, and identifies the organizational entities responsible for delivery of every element. At the top level of the tree is the solution itself; lower levels show how each element breaks down into sub-elements, etc. The breakdown must be complete at each level, including everything needed to integrate the elements to the next level. The depth of the tree is determined by the level of detail required for a reasonable cost estimate. At the lowest level, each element should be the responsibility of a single delivery organization, e.g. a team or service line, or a single subcontractor. The solution breakdown structure is adapted from the product breakdown structure as used in the PRINCE2 project management methodology [7].

B. *Solution-Based Estimating process*

Figure 1, taken from [5], shows the relationship between the various artifacts in our Solution-Based Estimation approach, and the roles responsible for their creation. A solution is designed to fulfill a business goal, which is elaborated into a set of functional and non-functional requirements. Based on these requirements, the solution architect designs a target solution and an architecture. Part of the documentation of this target solution's architecture is the solution breakdown structure (SBS). At the bottom of Figure 1, we see the project or delivery manager select an appropriate high-level delivery strategy for the solution, in close cooperation with the architect, using the target solution architecture as input. The delivery strategy, together with the SBS, are input for a work breakdown structure (WBS), which in turn leads to a project schedule. The elements in the SBS, the activities in the WBS and the schedule together form the basis for the estimates, resulting in a costing model for the solution.

III. DELIVERY BREAKDOWN VIEWPOINT

In order to use the architecture as input for estimation, it needs to be documented. A well-established good practice in documenting architectures is the use of multiple, concurrent views. This practice was introduced by Philippe Kruchten [8], and made its way into numerous other architecture documentation approaches and standards [9]. The idea behind the use of multiple views is that different stakeholders reading the architecture documentation have different concerns with respect to the system, and they each need to see how their specific concerns are addressed. A view is a partial description of the architecture that is specifically tailored to demonstrate how the architecture addresses one or more particular stakeholder concerns. A viewpoint is a template for constructing a particular type of view.

An essential part of the Solution-Based Estimating approach is the use of an architectural view that relates the solution architecture to the stakeholder concern of *costing the solution*. For reasons that will shortly become apparent, we call this the delivery breakdown view. The delivery breakdown view contains all the architectural information that serves as a basis for costing the solution, and consists of two constructs:

1. A solution breakdown structure, identifying the deliverable elements of the solution.
2. A high-level delivery strategy, describing how the various elements will be delivered as an integrated solution.

Typical stakeholders for the delivery breakdown viewpoint are (project/delivery/bid) management (for planning purposes), transition/transformation manager (for service solutions), estimators, cost engineers, purchasing staff and development teams.

The purpose of the delivery breakdown view is to document how the architecture addresses the overall stakeholder concern regarding costing: "what will be the cost of delivering the solution". This concern can be broken down into several sub-concerns, such as:

- What is the costing structure of the solution?

- How is the cost affected by individual solution requirements?
- What organizational entities are needed to deliver the solution (e.g. suppliers, service streams, subcontractors, development teams)?
- How can the solution be structured to meet a target budget?

A. Solution breakdown structure

A key aspect of architectural views is that they show how the solution can be decomposed from various perspectives in order to show how different stakeholder concerns are addressed. The solution breakdown structure, introduced in [5] and summarized in the Background section above, is the decomposition that frames the solution costing concern. It provides insight into the cost structure of delivering the solution by the following means:

- It breaks down the problem of estimating cost into smaller pieces: the deliverable elements whose individual delivery cost can be more easily and more accurately estimated than that of the complete solution.
- It allows breaking down the solution until the desired granularity level is achieved, i.e. the level of detail that is required for the current level of accuracy needed in the solution lifecycle: in early estimating, a high-level decomposition might suffice, and in later stages of the lifecycle the SBS may be further refined by adding decomposition levels.
- It decomposes a heterogeneous solution into homogeneous deliverable elements, allowing the use of sizing techniques for estimating homogeneous elements, such as functional size for software elements, square footage and power for hardware hosting, number of FTEs for organizational entities, or bandwidth, storage and computing capacity for infrastructure.
- When multiple parties are involved in the delivery, the SBS identifies the organizational entity that will deliver each element, so that it can be involved in costing its own part of the solution.

For each deliverable element in the solution breakdown, the view documents the requirements fulfilled by it, how and by whom it will be produced or delivered, and its cost driving properties. For enterprise-level solutions or services, these cost driving properties are heavily dependent on the type of the deliverable element: e.g., custom-made software cost can be driven by functional size [1], infrastructure elements by list price, services by monthly fees.

B. High-level delivery strategy

It is important to realize that, even though architecture plays an important role in cost estimation, the costing concern goes beyond the scope of architecture alone. As is apparent from the process depicted in Figure 1, delivery aspects play a crucial role. For this reason, any architectural viewpoint that addresses costing concerns should also frame delivery aspects. In the case of estimating the cost of delivering a project, delivery aspects like the choice of software development lifecycle should be part of the view. In the case of estimating the total cost of ownership (TCO) of a system or service solution, the choice of approach for transitioning the organization to the target solution should be considered as well, as should the strategy for operating the solution after the transition.

The delivery strategy in the delivery breakdown viewpoint is a high-level description of the development or acquisition approach for each element, plus an integration strategy that shows how all elements are integrated to deliver a single, coherent solution. Determining the delivery strategy is the responsibility of the project/(service) delivery manager, but she should consult the solution architect to make sure that all architectural concerns regarding the integration are addressed.

C. Discussion

To some architects, it may feel unnatural to "contaminate" architecture documentation with delivery aspects – aspects that they are usually not responsible for. As reasoned above, however, this is the only way to properly frame the costing concern. To further clarify this, let us explore the interdependence between system concerns and delivery concerns.

In the traditional separation of concerns between architect and (project/delivery) manager, the architect is the "content leader" [10], responsible for fulfilling the system requirements, and the manager is the "process leader", responsible for fulfilling delivery requirements (often called *constraints*). The architect makes architectural decisions to fulfill system requirements, the manager makes management decisions to fulfill delivery requirements. A too strict separation of these roles, however, does not take into account the effect that management decisions have on the fulfillment of system requirements, and the effect that architectural decisions have on fulfilling delivery requirements. Such "cross-context effects" can be detrimental, or beneficial if architect and manager involve each other to help fulfill requirements. For example, the architect can help fulfill delivery requirements by selecting technologies that the organization already has expertise in, or by decomposing the system into release packages to make delivery deadlines for features. Conversely, the manager can structure the development organization to match the architectural structure of the system, or implement review procedures to fulfill safety or security requirements.

Including the most significant delivery decisions in architecture documentation helps highlight these interdependencies. It puts the architecture in its delivery context, and promotes the necessary collaboration between architect and manager to fulfill both system and delivery requirements. The delivery breakdown viewpoint documents these combined system and delivery aspects to show stakeholders how their costing concerns are addressed.

The key is to limit the documented delivery choices in the architecture documentation to those which can be considered *architecturally significant.* A good rule of thumb for architectural significance of a design decision is its impact on stakeholders in terms of cost and risk [6]; the same rule of thumb can be used to identify the architecturally significant delivery decisions. Examples of delivery decisions that have high impact

are software development lifecycle choices, supplier selections and make-or-buy decisions – typically decisions in which architects should be involved, so it is not that far-fetched to include such delivery decisions in architecture documentation.

IV. EARLY FEEDBACK

We are currently in the process of validating the Solution-Based Estimating approach in various parts of our organization. Validation is done in two ways: by gathering feedback from teams that have applied the approach, and by post-mortem analysis of projects that have *not* applied the approach to see what the approach could have contributed.

Feedback from applying the approach was gathered by surveying 16 architects trained in RCDA that had applied Solution-Based Estimating in North-America and Europe. They were asked to indicate the impact of the approach on six aspects of estimating in the context of their most recent relevant project. The results are summarized in TABLE I. 5 respondents had not yet applied the approach. As is immediately apparent from the table, the vast majority of the architects that did apply it report positive impact for all aspects, with the exception of "ease to get approval from reviewers". This feedback encourages us to continue our initiative of applying architecture practices in cost estimating.

TABLE I. SURVEY RESULTS

Architect's assessment based on latest project	**Impact of Solution-Based Estimating**			
	Essential	***Helpful***	***No impact***	***Negative impact***
Completeness of estimate	4	7		
Accuracy of estimate	1	8	2	
Traceability of cost to solution	6	5		
Traceability from cost to NFRs	5	3	3	
Efficiecy of estimating process	1	6	4	
Ease to get approval from reviewers	1	5	4	1

This encouragement is further strengthened and even given a sense of urgency when we analyze projects that have not applied Solution-Based Estimating. We did such a post-mortem analysis for two projects. In both cases, by creation of a solution breakdown structure, we have identified gaps in the original estimates, mostly due to the fact that multiple organizational entities were involved, and without the presence of a delivery breakdown view, the organizational entities had interpreted the scope of their deliverable elements differently.

V. CONCLUSION

From the early feedback, it appears that using the architecture as a basis for costing is essential to achieve completeness of the cost estimate when multiple parties are involved in delivery of a heterogeneous, enterprise-level solution.

Combining the most significant technical and delivery aspects underpinning the cost estimate in one viewpoint creates overview for those stakeholders interested in cost-related concerns. This benefits the completeness and accuracy of the estimate, especially if many parties are involved. It not only helps place the architecture in context, but also enriches the architecture documentation with those parts of that context that are essential to understanding how the costing concerns are addressed.

This conclusion is subject to two important limitations. The first is that the early feedback was gathered from architects only. We have no reason to doubt their assessment of the impact of the Solution-Based Estimating approach on their projects, but the respondents should not be considered to be without bias towards an approach that promotes the importance of architecture in the organization. The second limitation is that validation has so far been within CGI only. The architects applying the approach have done so working for multiple clients in different market sectors and in various technology contexts and countries, which indicates at least some validity in other organizations; however, more research is needed before we can confidently state in which types of other organizations Solution-Based Estimating would be beneficial.

VI. REFERENCES

[1] ISO/IEC JTC1 SC7, *COSMIC International Standard (ISO/IEC 19761:2011),* ISO/IEC, 2011.

[2] B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer en B. Steece, Software Cost Estimation with COCOMO II, Englewood Cliffs, NJ: Prentice-Hall, 2000.

[3] M. B. Chrissis, M. Konrad en S. Shrum, CMMI, Guidelines for Process Integration and Product Improvement, Addison Wesley, 2003.

[4] R. Slot, A method for valuing architecture-based business transformation and measuring the value of solutions architecture (PhD Thesis), Amsterdam: University of Amsterdam, 2010.

[5] E. Poort en E. van der Vliet, „Estimating the Cost of Heterogeneous Solutions," in *Softw. Measurement and Intl Conf. on Softw. Process and Product Measurement (IWSM-MENSURA), 2014 Joint Conf. of Intnl Workshop on*, Rotterdam, 2014.

[6] E. R. Poort en H. van Vliet, „RCDA: Architecting as a Risk- and Cost Management Discipline," *Journal of Systems and Software,* pp. 1995-2013, 2012.

[7] Office of Government Commerce, Managing Successful Projects with PRINCE2, TSO, 2009.

[8] P. Kruchten, „Architectural Blueprints—The "4+1" View Model of Software Architecture," *IEEE Software,* pp. 92-95, November 1995.

[9] ISO/IEC JTC1 SC7, *Systems and software engineering - architecture description (ISO/IEC 42010:2011),* 2011.

[10] G. Muller, „Architect as Content Leader," 2 February 2014. [Online]: http://www.gaudisite.nl/ArchitectAsContentLeaderSlides.pdf. [Retrieved 12 January 2015].

doi: 10.1109/WICSA.2015.9